This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problems Mailbox.

THIS PAGE BLANK (USPTO)

EUROGRAPHICS 2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19 (2000), Number 3

Generating Animatable 3D Virtual Humans from Photographs

XP-002257872

P.D. 21-08-2000
P. C1-10
10

WonSook. Lee, Jin Gu, and Nadia Magnenat-Thalmann

MIRALab, CUI, University of Geneva, Switzerland
Web: http://www.miralab.unige.ch
E-mail: {wslee, gu, thalmann}@cui.unige.ch

Abstract

We present an easy, practical and efficient full body cloning methodology. This system utilizes photos taken from the front, side and back of a person in any given imaging environment without requiring a special background or a controlled illuminating condition. A seamless generic body specified in the VRML H-Anim 1.1 format is used to generate an individualized virtual human. The system is composed of two major components: face-cloning and body-cloning. The face-cloning component uses feature points on front and side images and then applies DFFD for shape modification. Next a fully automatic seamless texture mapping is generated for 360° coloring on a 3D polygonal model. The body-cloning component has two steps: (i) feature points specification, which enables automatic silhouette detection in an arbitrary background (ii) two-stage body modification by using feature points and body silhouette respectively. The final integrated human model has photo-realistic animatable face, hands, feet and body. The result can be visualized in any VRML compliant browser.

1. Introduction

In recent years, modeling virtual human body has attracted more and more attention from both the research and industrial community. It is no longer fantasy to imagine that one can see herself/himself in a virtual environment moving, talking and interacting with other virtual figures or even with real humans. By advances in algorithms and new developments in the supporting hardware this fantasy has become a reality.

The issues involved in modeling a virtual human model are as follows:

- acquisition of human face and body shape data
- realistic high-resolution texture
- functional information for animation of the human face and body

We address how to acquire an animatable human body with a realistic appearance. It is our goal to develop a technique that enables an easy acquisition of the avatar model having the ability to be animated well and produced at a low cost. There are two basic types of techniques for obtaining 3D object models, according to the different requirements for the models. The first type of technique focuses on the accuracy and precision of the obtained object models, such as those used in CAD systems and industrial applications. The second type of techniques concentrates on the shape and visual realism of the reconstructed models, such as those used in virtual reality applications.

When we have to place importance on the accuracy of the shape, there are various approaches to the reconstruction of a face either using a sculptor[8], a laser scanner[22], a stereoscopic camera[21], an active light stripper[24], video stream[16,9]. In recent years, body cloning has also become an increasingly hot topic. Similarly, there are many methods that concern precision and accuracy[16, 11, 1, 7, 10, 13]. Generally, these systems are either expensive or require expertise knowledge in using them and need a special environment setting. Thus, most of them have limitations when compared practically to a commercial product (such as a camera) for the input of data for reconstruction and finally animation.

On the other hand, systems using the second type of techniques are much cheaper and easier to use. These techniques are usually model-based. There are several approaches to

© The Eurographics Association and Blackwell Publishers 2000. Published by Blackwell Publishers, 108 Cowley Road, Oxford OX4 1JF, UK and 350 Main Street, Malden, MA 02148, USA.

IOCID: <XP___2257872A__1_>

the reconstruction of either a face[2, 15, 17, 20] or a body[12] from photo date. These approaches concern mainly the individualized shape and visual realism using a high quality image input. For example, Hilton et al.[12] proposed a method for cloning virtual people. A generic human model is taken and information extracted from photos is used to modify the generic model. The approach is simple and efficient. However this method does not give a good reconstruction and animation for the face. In addition, their generic model is not seamless, which means the regions around certain skeleton joints do not have the smoothly connected surface, so that the final textured model has some mismatching problem when we animate the joints. It also lacks the flexibility in terms of the imaging environment since it requires a specially prepared background and properly controlled lighting when images are taken.

Our approach, which belongs to the second type, addresses the following questions and suggests the solutions.

What to produce from what?

We produce a realistic and animatable whole body including the face, hands and body from photo data. Every body parts are smoothly connected and textured. Photographs of the whole body cannot provide sufficient facial information in order to construct a good face model and further facial animation. Therefore, we take two additional photos that focus on the face only, besides the three whole body photos.

How easy is the environment to get the input?

We use simple snapshots with commercial cameras without any special environment. Instead of seeking a solution by using special environment, we provide a user-friendly interface, which allows non-expert user to interactively hint to the system certain important information about the human body. In this way, with a little amount of user interaction, we achieve more flexibility in using the system.

How automatic is the processing for users?

We provide an automatic system except for a few interactions at the beginning as shown in Figure 1.

How much can we animate?

The individualized virtual human inherites the functional structure from the generic human with animation capacity on the face and body.

How easy is it to visualize with other applications?

The VRML Humanoid Animation Working Group (H-Anim) exists for the main purpose of creating a standard VRML representation for humanoid. Our generic body is in VRML H-Anim 1.1 format[14] and the resulting body can be visualized by web browsers, such as Netscape and animated by a JAVA program.

The outline of the algorithm is shown in Figure 1. Section

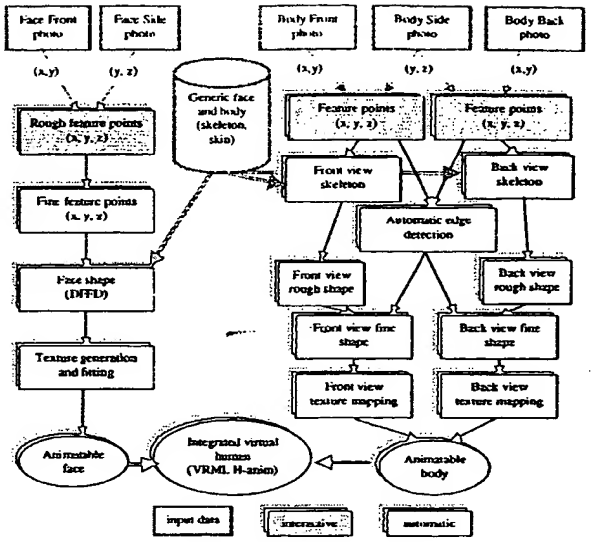

Figure 1: *Overflow of face and body cloning*

2 is devoted to the face-cloning program while Section 3 explains the body cloning. The results are shown in Section 4 and are concluded in Section 5.

2. Face cloning

2.1. Shape modeling

In this section, we present a way to reconstruct a photo-realistic head for animation from orthogonal pictures. First, we prepare a generic head with an animation structure and two orthogonal pictures of the front and side views. The generic head has efficient triangulation, with finer triangles over the highly curved and/or highly articulated regions of the face and larger triangles elsewhere. It also includes eyeballs and teeth.

The main idea to get an individualized head, is to detect feature points (eyes, nose, lips, and so on) on the two images and then obtain the 3D position of the feature points to modify a generic head using a geometrical deformation. The feature detection is processed in a semi-automatic way. The user sets a very few feature points (key points) and the other feature points are fitted using a *piecewise affine transformation* first and then snake methods. The structure snake method with some anchor functionality is described in another paper[19]. Then, two 2D position coordinates in the front and side views, which are the XY and the ZY planes, are combined to be a 3D point. After using a global transformation to move the 3D feature points to the space for a generic head, Dirichlet Free Form Deformations (DFFD)[23] are used to get new geometrical coordinates of a generic head adapting to the detected feature points. The control points for the DFFD

© The Eurographics Association and Blackwell Publishers 2000.

ID: <XP 2257872A I >

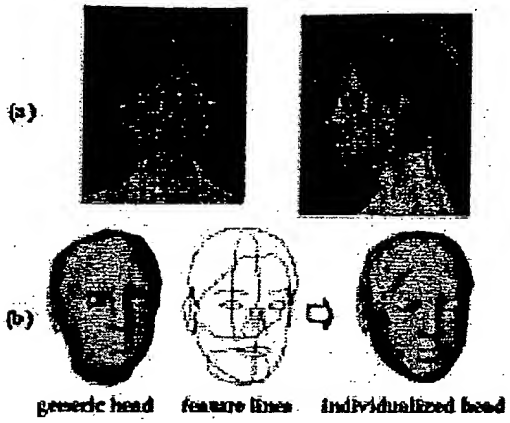

Figure 2: *(a) normalization and features. (b) Modification of a generic head with feature points*

are feature points detected from the images. Then the shapes of the eyes and teeth are separately adapted to the new head with translation and scaling from the generic model. Figure 2 shows the steps for head modification from photos.

2.2. Texture mapping

Texture mapping is useful not only to cover the rough matched shape, as here the shape is obtained only by feature point matching, but also to get a more realistic colorful face.

The main idea of texture mapping is to get an image by combining two orthogonal pictures in a proper way to get the highest resolution for the most detailed parts. The detected feature points data is used for automatic texture generation by combining two views (actually three views by creating the left view by flipping the right view). We first connect two pictures with a predefined index for feature lines using a geometrical deformation (see Figure 3 (a)) and a multiresolution technique[6] for removing boundaries between different image source (see Figure 3(b)). The eyes and teeth images are added automatically on top of an image, and these are necessary for the animation of the eyes and mouth region.

To give a proper coordinate on a combined image for every point on a head, we first project an individualized 3D head onto three planes such as the front (XY), right (ZY) and left (ZY) directions. With the information of the predefined index for feature lines, which are used for image merging above, we decide on which plane a point on a 3D head is projected. Then projected points on one of three planes are transferred to either the front feature points space or the side feature points space in 2D. Finally, a transform on the image space is processed to obtain the texture coordinates. More details are found in the paper[20].

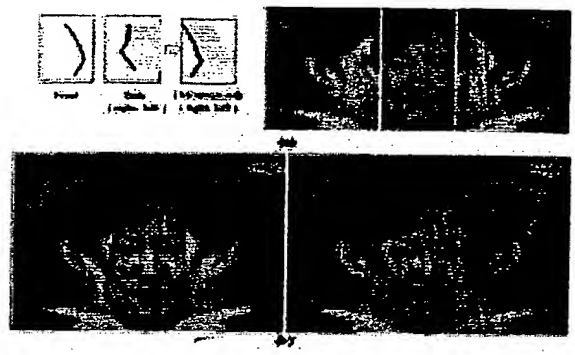

Figure 3: *(a) A geometrical deformation for the side views to connect to the front view (b) before and after multiresolution techniques.*

Figure 4* shows several views of the final reconstructed head out of two pictures in Figure 2(a). When we connect this head with a body, we remove the neck (see the second last face in Figure 4*) since the neck is from the body due to the body skeleton animation for face rotation. The face animation is immediately possible as being inherited from the generic head as shown in the last face in Figure 4*.

Figure 4: *snapshots of a reconstructed head in several views and animation on the face*

3. Body cloning

Our body cloning is a model-based method. We use two main inputs. The first input is the generic body. The second is still photos of a person to be cloned. We assume the person wears trousers and not too loose clothes. We deform the generic body to adapt to the individualized body.

3.1. Generic body structure

The generic body is in MPEG-4 compatible H-Anim 1.1 formats[14]. The skeleton and several skin parts displayed with several colors are shown in Figure 5 where the skin parts are smoothly connected. It has 94 skeleton joints and 15 skin parts including *head*, *right_hand*, *left_hand*, *right_foot* and *left_foot*. The first version of generic body we are using is collected from a public domain[3] and modified for our usage. Each skin part is saved in local coordinates and is related to a skeleton joint as shown in Figure 6, where the skeleton location is indicated by arrows and related skin parts are

© The Eurographics Association and Blackwell Publishers 2000.

OCID: <XP 2257872A >

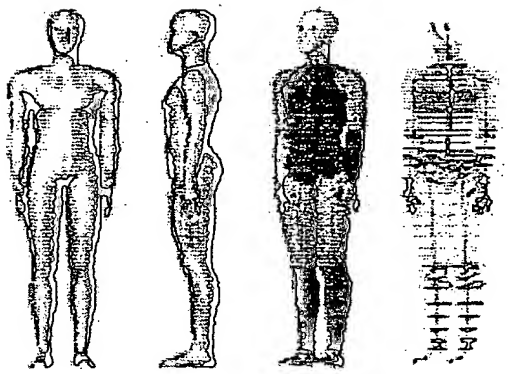

Figure 5: *The seamless generic body with H-Anim 1.1 skeleton and several skin parts*

written inside (). The right side skin parts are not shown, but it is easy to guess from the left side. Each skin part is transformed into global coordinates by a 4x4 matrix which connects to the corresponding skeleton joints.

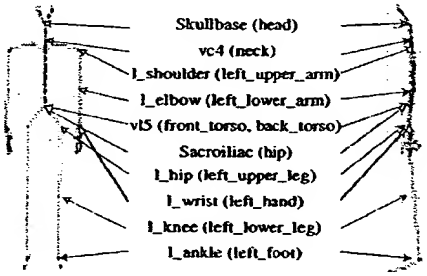

Figure 6: *The H-Anim joints related to skin parts*

The 12 skin parts beside *head*, *hands* and *feet* are designed to have real-time deformation for animation[3]. The good points of these skin parts are that first they compose a seamless skin envelope, so that the texture mapping will be smoothly connected with animation. Secondly the way how to organize the points is specially designed such that each skin part is composed of several slices and each slice in the skin part has the same number of points. For example the *hip* has 6 slices and 26 points on each slice (the total point number on *hip* is 6x26 = 156). We call it as the *grid structure*, which makes the *piecewise affine transformation* possible in later sections.

The *head*, *hands* and *feet* are separate objects with different structures from *grid structure*. They are also animated in different ways.

3.2. Taking photographs and initialization

We focus on the simplest environment to take photos with only one camera. We take three photos, from the front, the side and back. In this case, the front and back views are not exact reflections of each other since we asked the person to rotate for the back view after taking the front view.

We input the height of the person and the image body heights are checked on the three images for normalization. Figure 7 shows normalized images. Since we use arbitrary background to take photos and the size of the person is not fixed, we use interactive feature points localization on images as shown as small points in Figure 7. This simple feature points localization is used for skeleton modification, rough skin deformation and automatic edge detection in the next sections.

Figure 7: *Feature points on three images*

3.3. Skeleton modification

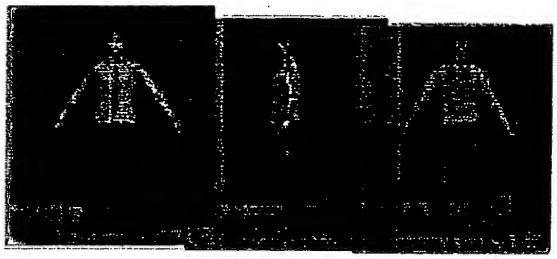

Figure 8: *Automatic Skeleton fitting with feature points.*

When we have feature points for the skin envelope (even though the person put on clothes, we assume that the clothes outlines are close to the skin outlines), we can have an estimation of the skeleton. For example, for the *r_elbow* must be located around middle position between the right end shoulder point and outer end point of the right wrist. Here we apply affine transformations and Barycentric interpolation to find the typical skeleton joints from feature points. Since there are 94 skeleton joints, we make a subset of joints as key joints, which are modified by feature points while others

© The Eurographics Association and Blackwell Publishers 2000.

ID: <XP 2257872A 1 >

are modified by *piecewise affine transformations* defined by key joints and the skeleton hierarchy.

The skeleton joints of the *head*, *hands* and *feet* are simply scaled and translated with the transformation between the generic body's skeleton and the person's skeleton, for example *vt1* and *HumanoidRoot* can be used to find the transformation. Figure 8 shows the skeleton modified by the front and the side views. Since the front and back views do not have the exactly same pose, the skeleton modified by the front view does not match on the back view.

3.4. Rough skin modification

As we mentioned in the section 3.1, each skin part is connected to a skeleton joint by a 4x4 matrix to be in global coordinates. We update the matrix by scaling, translation and rotation defined by the corresponding skeleton joint and the child skeleton joints. As we see in Figure 9, adjacent skin parts are not guaranteed to be continuous. The shoulder parts are overlapping with torso parts.

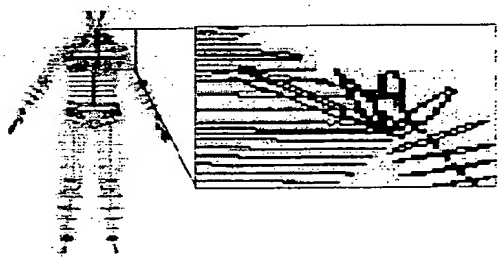

Figure 9: *Updated skin parts by skeleton joints and related matrix*

A simple linear transformation by 4x4 matrix does not solve the overlapping or separating problems. So we need a special transformation to solve the overlapping (or separating) problem and integrate the skin parts properly with an approximated shape to the person.

Here we define a freeform deformation to make a rough matched continuous body with feature points information. The control points are placed at certain required positions to represent the shape characteristics. Hence the skin model can be deformed by moving these control points, which is designed such as they have corresponding points on the front (or back) and the side view images, or the locations are found with certain relations. For example the boundary between the *front_torso* and *right_upper_arm* can be found from the feature points on the bottom-right corner point of the neck and on the right end shoulder point on the images. Furthermore, several control points are located at the boundaries between two parts, so that surface continuity is preserved when the posture of the generic body is changed. These control points are used for the *piecewise affine transformation*.

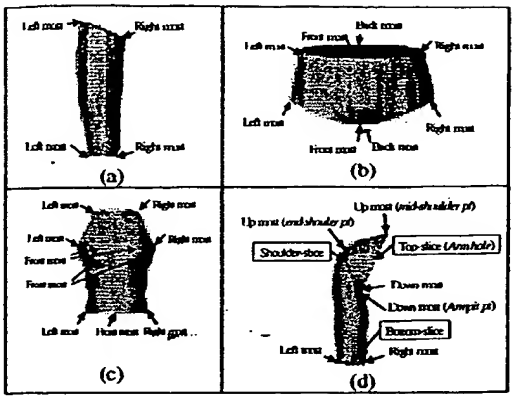

Figure 10: *The control points on right_upper_leg, hip, front_torso and right_upper_arm parts*

We apply separate deformations for each skin part. We show some examples of the control points in Figure 10.

We apply first *piecewise affine transformation* where each affine transformation is defined on each segment on a curve. Thanks to the *grid structure* of skin parts, we apply the *piecewise affine transformation* horizontally first on slices with control points and then vertically on points which have the same index on each slice. For example for the *hip* part, we define the first affine transformation with the left-most point on the top-slice and the front-most point on the top-slice and corresponding control points on images. Also we apply the affine transformation on points between the left-most point and the front-most point. Then we define the next affine transformation for front-most point and right-most point and apply it to points in between. We define and apply the third and forth affine transformations on the other two pieces on the top-slice. Then we get a new top-slice from the generic body's slice, which fits to the person's top-slice now. Then we apply the similar process for the bottom-slice that has control points too. After getting the new bottom-slice, we define an affine transformation in the vertical direction using two points with the same index on the top-slice and the bottom-slice. Figure 11 shows the steps for the *hip* part. For the **_upper_arm* parts, we apply *piecewise affine transformation* for three slices such as the top-slice, the shoulder-slice, and the bottom-slice as shown in in Figure 10 (d).

Figure 11: *The process of the affine transformations for the hip*

© The Eurographics Association and Blackwell Publishers 2000.

After applying the *piecewise affine transformation*, we make one more process to stick adjacent skin parts properly. When the generic body is loaded, the connection database is built automatically and it is used for the skin parts connection. The database contains which point on which skin part is connected to which point on which skin part. The *neck* is not deformed using the feature points from images since it is too small on the images. The *piecewise affine transformations* are defined from adjacent points on the *head* with the top-slice on *neck* and from adjacent points on the *front_torso* and *back_torso* with the bottom-slice on *neck*.

It is a rough deformation since we deform the generic body only with a few feature points. So it does not match exactly for the outfit on the images. However it serves as an initialization of skin for the shape and catches a proper functional approximation for animation such as having a proper skeleton and skin parts localization.

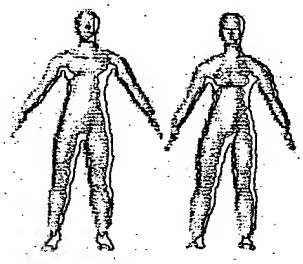

Figure 12: *two bodies (front+side/back+side) with rough skin deformation*

Since the front and back views were taken in different positions, we produce two bodies as seen in Figure 12. The left one obtained from the front and the side images and the right body from the back and the side images. The reason to produce two bodies is because of two sources for texture mapping, which will be described later.

3.5. Heuristic based silhouette extraction

There are plenty of literatures available about boundary extraction or edge linking[5, 18]. It can be treated as a graph-searching problem, as an optimization problem, or as an energy minimization and regularization procedure. However, these algorithms are usually inefficient due to the need for backtracking or exhaustive search. Or the algorithms need time to reach convergence or stable result, such as the snake algorithm. We design a simple algorithm by making use of the feature points on the body. In this section these feature points serves as the heuristics for the body silhouette extraction.

First, Canny edge detector is applied to every image. Then a coloring-like linking algorithm is used to link the edgels (edge pixels) into connected segments. Due to possible noise caused by the background, the edgels generated by the background sometimes are connected to the body edgels. To avoid this potential wrong connection from occurring, we split the segments into short line segments.

To make the following discussion easier, let us call the line segments formed by consecutive feature points as *feature segments*, while the short line segments formed by linking edge pixels, as *edge segments*. Each feature segment indicates the vicinity and approximate direction of the boundary to be found. From the Canny edge detector and linking step, we obtain the *edge segments* generated by the object as well as by the background. We first throw away those lying outside the vicinity of any *feature segments*. The goal now is, for each feature segment, to find a path that is formed by an ordered set of *edge segments* within its vicinity.

To link the *edge segments* into meaningful boundaries, we first look for the admissible connection for each edge segment. See Figure 13. We define a connection between edge segment $S_1 = P_{12} - P_{11}$ and $S_2 = P_{22} - P_{21}$ as admissible if:

$$S_1 \cdot S_2 \geq 0.0, \quad \alpha_1 < T_{sm}, \quad \alpha_2 < T_{sm}$$

where P_{11}, P_{12}, P_{21} and P_{22} are the ends of the two segments under consideration, α_1 is the angle between S_1 and the potential connecting segment $C_{12} = P_{21} - P_{12}$, α_2 is the angle between C_{12} and S_2, T_{sm} is the maximum angle allowed for the connection. Next in order to select the most desired connection, we define a function G^L to evaluate the "goodness" of the potential connection :

$$G^L(S_1, S_2) = cos(\alpha_1)cos(\alpha_2) \frac{(1 + cos(\alpha_1)cos(\alpha_2)log(M_2))}{(wD_1 + (1-w)D_2)}.$$

where D_1 is the length of the C_{12} and D_2 is the distance from the P_{22} to the feature segment L, M_2 is the edge magnitude of edge segment S_2, w is the weight of D_1 taken as a constant in our experiments. The rationale behind this definition is that we always favor a connection that contributes to a smooth path running along L, formed by segments with strong edge magnitude.

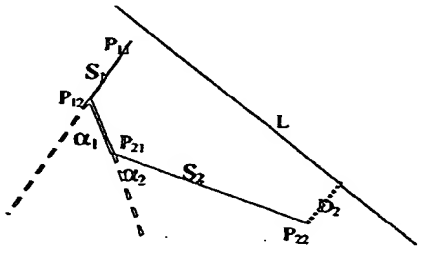

Figure 13: *Link two edge segments*

Thus based on G^L, we find, for the two ends of each edge

© The Eurographics Association and Blackwell Publishers 2000.

XP__2257872A__I_>

segment, its best connection that maximizes G^L among all of its neighboring *edge segments*. Now we are ready to build the path. Starting from each edge segment, we connect it to its two best connections computed by G^L to form a partial path. For the *edge segments* sitting on the head and tail of this partial path, we connect their open ends to their respective best connections. This procedure is repeated until there is no further connection possible. So a path $\mathbf{P}$ consists of a sequence of *edge segments* S_i, $i = 1,2,...,N$. Now we need another evaluation function to assess the "goodness" of a path. We define a function G^P as:

$$G^P(P) = \Sigma M_i / (w_1 DT_1 + w_2 DT_2 + w_3 \Sigma D_i (1 + sin(\alpha_{i,1}) + sin(\alpha_{i,2})))$$

where ΣM_i is the summation of the edge magnitudes, DT_1 is the distance between the path ends to the ends of the feature segment, and DT_2 is the average distance from the pixels on the path to the feature segment. D_i, $\alpha_{i,1}$ and $\alpha_{i,2}$ correspond to D_1, α_1 and α_2 in Figure 13, respectively. w_1, w_2, w_3 are the weights of each measurement (here we take the value 0.2, 0.2 and 0.8 respectively). Among all the paths found, the one $\mathbf{P}^*$ maximizing G^P is selected, i.e. $\mathbf{P}^* = arg\max_P(G^P(\mathbf{P}))$. We show a few examples in Figure 14.

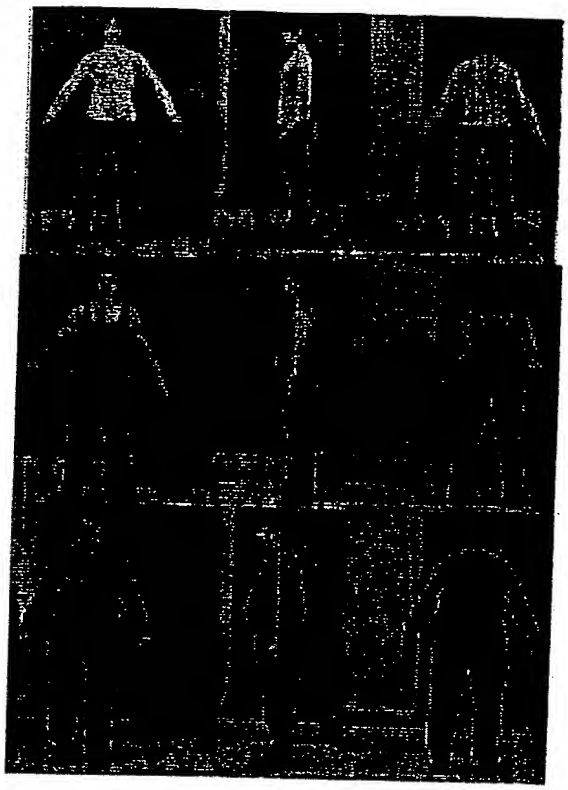

Figure 14: *Images with super-imposed silhouette*

3.6. Fine skin modification with silhouette information

After the skeleton adjustment and rough skin modification in the previous sections, the skin parts have been adjusted into proper orientation and rough size. Now we discuss how the image silhouettes are used to further modify the body so that the final body will produce the same silhouettes as those extracted from the images.

To build the 2D-3D association, we back project the 2D into 3D space. The silhouettes from the front or back view are mapped onto the XY plane and also the side view is mapped onto the ZY plane. For each view v, every slice S_i has two points, V_{i1}^v and V_{i2}^v on the occluding slice. These two points correspond to two pixels on the silhouette. Denote these two corresponding pixels as P_{j1}^v and P_{j2}^v. Generally the number of pixels is much larger than the number of slices, so we use the following formula to select the proper pixel pair:

$$P_{jk}^v = P_1^v + (MC_i - MC_1)/(MC_K - MC_1)(P_N^v - P_1^v)$$

where $k = 1,2$, and P_1^v and P_N^v are the first and last pixel on the silhouette, $MC_i, i = 1,2,...,K$ is the mass center of slice i. All the parts except the arms have silhouettes from two views, i.e. front/back and side views. The arms only have silhouettes extracted from the front/back view. Now the problem is reduced to how to modify the model slice given 2 or 4 data points that are associated to points on the slice. We first apply a global translation to all the points on the slice so that the mass center of the slice coincides with the midpoint of the two back-projected pixels. Then we do a global scaling to the slice. The scaling factor is estimated as the ratio of the distance between the two associated pixels and that between the two points. The silhouette from front/back image is used to scale the slice on XY plane and that from side view is used to scale in ZY plane. In order to ensure the two points V_{i1}^v and V_{i2}^v that sit on the occluding slice to produce the pixels same as the two associated pixels P_{i1}^v and P_{i2}^v, we apply a translation $T_{i,m}$ to each point $V_{i,m}$ of the slice as follows:

$$T_{i,m} = w(P_{i1}^v - V_{i1}^v) + (1 - w)(P_{i2}^v - V_{i2}^v)$$

where $w = ArcL(V_{i,m}, V_{i2}^v)/ArcL(V_{i1}^v, V_{i2}^v)$, $m = 1,2,...,N_i, N_i$ is the number of points on slice S_i, and $ArcL(.)$ is a function computing the arc length of the slice curve. This makes sure the modified slice will generate the exactly same image pixel points under the same projection while keeping the curve smoothness. Special care is needed for the shoulder-upper arms joining. The generic body has slices that are used to smoothly transit the shoulder to upper arm (see Figure 10 (d)). However there is actually no explicit corresponding information in the image. Fortunately, we can rely on the association of the upper arm with the torso to adjust these few slices. First of all, the orientation and the

© The Eurographics Association and Blackwell Publishers 2000.

CID: <XP 2257R72A

size in XY plane of these slices are adjusted by making use of the silhouette generated by the *middle shoulder point* and the estimated *armpit point*. Secondly, we have to enforce the upper arm to be seamed to the armhole, which is associated with the torso. In such a way, we can estimate a rough scaling in YZ plane since the torso has been modified by its side view silhouettes.

3.7. Texture mapping

We use only the front and back views for texture mapping since the two views are enough to cover the whole body except for the head. The head texture mapping is done with the front and the side views as shown in the section 3.2.

For the texture mapping, we have to give texture coordinates to points on skin envelope. Since there are two images used, we have to make a partition of the skin envelope polygons, either to the front view or to the back view by checking the cross product of the vertex normal with the viewing vector. If the point belongs to a front (back) view polygon (i.e. visible to front/back view point), we define the point as having front (back) view. If the vertex belongs to both a front view polygon and a back view polygon, we define the vertex as having front+back view. Since the vertex with front+back view is located on the boundary of the front and side views, we set two texture coordinates, one in the front view image and the other in the back view image. For the other vertices, it is straightforward to set the texture coordinate either in the front view image or in the back view image.

To get the texture coordinates, we use a projection onto the XY plane in the image space. Here we have to pay attention since there are two bodies either from the front and the side views or the back and side views. We follow the process such as:

1. deform the body with the back and side views;
2. project back/front+back viewpoints onto the back view image plane to get the texture coordinates;
3. deform the body with the front and side views;
4. project front/front+back viewpoints onto the front view image plane to get the texture coordinates.

Then the final individualized body has the proper texture mapping on both the front view and the back view. However there are still some problems on the boundaries as shown in the left side images in Figure 15. There are mainly two reasons causing this problem. First, due to the size of polygons on the virtual body, which is much bigger than the pixel size of images, the projected boundaries of the triangles on the frontal and back view boundary do not match to the detected boundaries based on pixel size. In addition due to the limited digitization resolution of the camera, the pixel colors on the boundary of foreground and background are usually the smeared combination of the boundary and foreground color. Furthermore the noisy effect of the texture is magnified by the 3D triangles on the boundaries. Secondly although the two images used for texture mapping are usually taken under same illumination condition, they are not necessarily to have the same visual intensities or colors. So when they are mapped onto the 3D body, the difference in the color and intensity can be easily perceived.

In order to remove the first cause by digitization process, we modify the pixel colors within the neighborhood of each edge pixel. Since from the previous edge detection processing, we have already known which side of the boundary is background, we search along the perpendicular direction to the edge for a foreground pixel and take its color as the color for the edge neighborhood pixels. As result shows, this simple processing removes the noisy effect of the digitization process.

Next, we need to smooth the frontal and back texture to remove the difference between the two images. From the feature points given in the previous processing, we can recognize semantically the various body parts, hence establish the part correspondences between the two images. We further find the pixel correspondence according to the boundary lengths. Within the neighborhood of the two pixels from frontal and back view images respectively, we use a linear blending. Let C^F and C^B are edge pixels in correspondence from the frontal and back view images respectively. Then for any pixel p^F and p^B in the linear neighborhood of length L_ϵ defined to be perpendicular to the local boundary at C^F and C^B respectively, we compute its color using the following blending function:

$$F^{bld}(p^F) = \alpha_1^F F(p^F) + (1.0 - \alpha_1^F) B(p^B)$$

$$B^{bld}(p^B) = \alpha_1^B F(p^F) + (1.0 - \alpha_1^B) B(p^B)$$

where $\alpha_1^F = 0.5(1.0 + d(C^F, p^F)/L_\epsilon)$, $\alpha_1^B = 0.5(1.0 + d(C^B, p^B)/L_\epsilon)$, $F(.)$ and $B(.)$ denote the original color of the original p^F and p^B while $F^{bld}(.)$ and $B^{bld}(.)$ denote the blended color for the pixels under consideration.

Figure 15 shows the texture mapping results before and after texture blending.

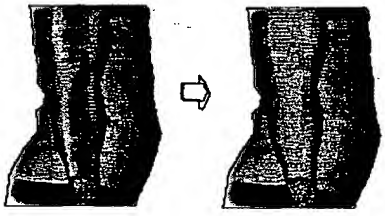

Figure 15: *The effect of texture blending*

4. Connection between body and face and results

We processed face cloning and body cloning separately. Even though the size of the face is much smaller, we have

© The Eurographics Association and Blackwell Publishers 2000.

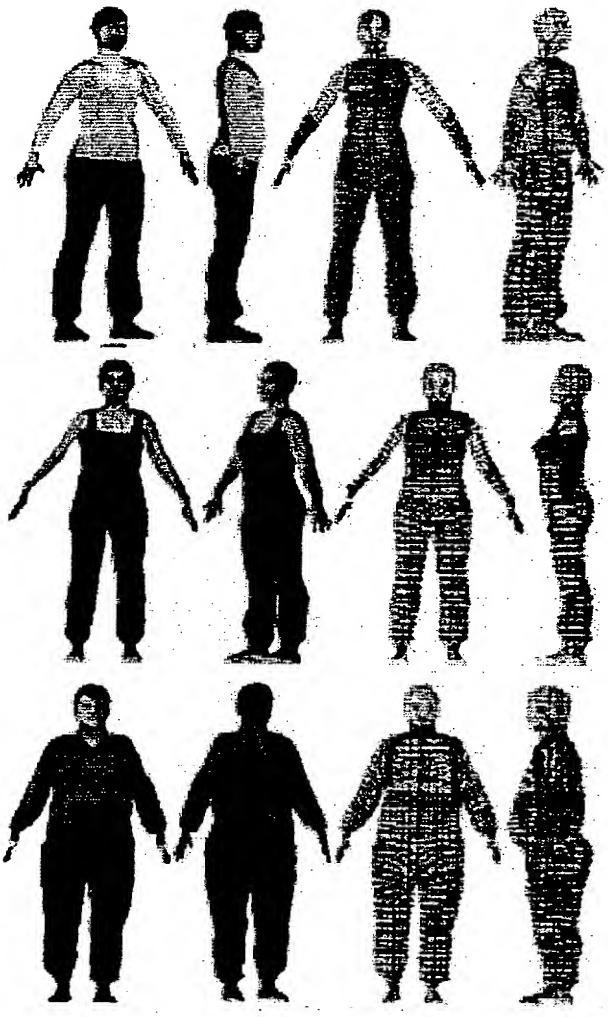

Figure 16: *Final H-Anim 1.1 bodies with detailed individualized faces connected to the bodies. They are modified from one generic model*

to keep a detailed structure and high resolution for a face since we often zoom in the face to see facial animation and communication. So we use separated images for face cloning and body cloning and these two cloning methods use different texture mapping schemes. The body texture comes from the front and back view while the face texture comes from the front and side view due to the sphere like shape, which needs a side view for proper texture for ear parts. When we have a face and a body reconstructed separately, we have to connect them properly to make a smooth envelope for a perfectly smoothed body. We check the face size and location of the face on the front and side view body images using feature points as shown in Figure 7. Then simple translation and scaling locate the individualized face on the individualized body. We use an automatic sticking between them by finding the nearest points on the neck and the head to ensure the connection.

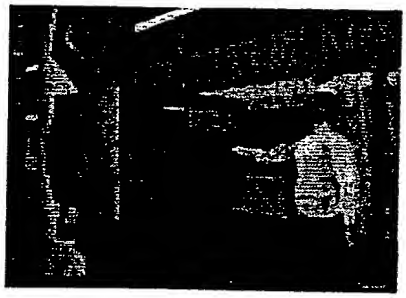

Figure 17: *Animation in a virtual environment*

The final bodies are shown in Figure 16* with the input images in Figure 14. Since the generic body had H-Anim 1.1 structure, the individualized bodies keep the same structure in VRML H-Anim 1.1 format, which means we can animate the bodies immediately. The final bodies are exported into VRML format and can be loaded in public web browsers. Figure 17 shows an animation example in a virtual environment[4]. Since we are using seamless skin structure for a real-time animation, the skin and textures are smoothly connected during animation.

5. Conclusion

In this paper, we introduce a model-based approach to photo-realistic animatable virtual human cloning from several pictures. The method takes as input two photos of the face of the subject and three photos of her/his body. Two different cloning methods are employed to face and body respectively. The efficient and robust face cloning method shows the processes of modifying a generic head for shape acquisition and producing texture images by combining orthogonal image pair smoothly. An H-Anim 1.1 compliant generic body is taken to serve as our reference model for a body. Unlike other existing systems, which require special environment to obtain input data, we seek the solution through a friendly user interface for an accurate localization of feature points, which are used both for automatic edge extraction and for modifying the generic body into individualized ones. A simple but effective heuristics-based boundary extraction algorithm is proposed to automatically extract the body silhouette from the images. Then to avoid the possible overlapping for skin parts, we introduce a two-step modification, first rough matching just with feature points information and then fine matching with detected edge information. The body texture mapping is processed using two images. Moreover, we connect the individualized head to the

© The Eurographics Association and Blackwell Publishers 2000.

individualized body to form a complete animatable human model. As a result, we are able to animate the cloned human models in virtual environments. This method shows better cloning of the whole body in terms of both reconstruction and animation. The robustness and practical usefulness of the face reconstruction method is proved on several public demonstrations such as ORBIT'98 in Basel (CH), CEBIT'99 in Hannover (DE), SMAU'99 in Milano (IT), and TELECOM'99 in Geneva (CH). In these events, hundreds of people (Asian/Caucasian/Africa, female/male, young/old) were cloned and animated in a virtual world in around 5 minutes. The whole body reconstruction also takes similar time.

The automatic reconstruction of 3D clothes from the same photo input as we use here is the ongoing research topic.

Acknowledgment

The authors would like to thank Laurent Moccozet, Prithweesh De, and Christian Babski for their help. We are grateful to people taken photos including eRENA partners. This project is supported by Swiss National Research Foundation and partially supported by Hong Kong University of Science and Technology.

References

1. S. Addleman. Whole-body 3d scanner and scan data report. In *Three Dimensional Image Capture*, pages 2–5. 1997.
2. T. Akimoto, Y. Suenaga, and R. S. Wallace. Automatic creation of 3d facial models. In *IEEE Computer Graphics And Applications*. 1993.
3. C. Babski and D. Thalmann. A seamless shape for hanim compliant bodies. In *Proc. VRML 99,ACM Press*, pages 21–28, 1999.
4. C. Babski and D. Thalmann. Realtime animation and motion capture in web human director(whd). In *Proc. Web3D And VRML2000 Symposium*, 2000.
5. D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall InC, 1982.
6. P. J. Burt and E. H. Andelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2(4):217–236, 1983.
7. H. Daanen, S. E. Talory, M. A. Brunsman, and J. H. Nurre. Absolute accuracy of the cyberware wb4 whole body scanner. In *Three Dimensional Image Capture*, pages 6–12. 1997.
8. T. DeRose, M. Kass, and T. Truong. Subdivision surfaces in character animation. In *Proc. SIGGRAPH 1998*, pages 85–94, 1998.
9. P. Fua. Face models from uncalibrated video sequences. In *Captech98 (Modelling and Motion Capture Techniques for Virtual Environments)*, pages 215–228, 1998.
10. P. Fua. Human modeling from video sequence. *Geomatics Info Magazine*, 13(7):63–65, 1999.
11. J. Gu, T. Chang, S. Gopalsamy, and H. Shen. A 3d reconstruction system for human body modeling. In *Captech98 (Modelling and Motion Capture Techniques for Virtual Environments)*, pages 229–241, 1998.
12. A. Hilton, D. Beresford, T. Gentils, R. Smith, and W. Sun. Virtual people: Capturing human models to populate virtual worlds. In *Proc. Computer Animation 1999*, pages 174–185, 1999.
13. http://www.cyberware.com.
14. http://www.H-Anim.org.
15. H. S. Ip and L. Yin. Constructing a 3d individual head model from two orthogonal views. *Visual Computer*, 12:254–266, 1996.
16. I. A. Kakadiaris and D. Metaxas. Human body acquisition from multiple views. In *Proceedings of the Fifth ICCV 1995*, pages 618–623, 1995.
17. T. Kurihara and K. Arai. A transformation method for modeling and animation of the human face from photographs. In *Proceeding of Computer Animation*, pages 45–58, 1991.
18. S. F. Lai and R. T. Chin. Deformable contours: Modeling and extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1084–1090, 1995.
19. W.-S. Lee, P. Kalra, and N. Magenat Thalmann. Model based face reconstruction for animation. In *Proc. Multimedia Modeling (MMM)*, pages 323–338, 1997.
20. W.-S. Lee and N. Magnenat-Thalmann. Head modeling from pictures and morphing in 3d with image metamorphosis based on triangulation. In *Captech98 (Modelling and Motion Capture Techniques for Virtual Environments)*, pages 254–267, 1998.
21. W.-S. Lee and N. Magnenat-Thalmann. Fast head modeling for animation. *Image and Vision Computing*, 18(4):355–364, 2000.
22. Y. Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. In *Proc. SIGGRAPH 1996*, pages 55–62, 1996.
23. L. Moccozet and N. Magnenat-Thalmann. Dirichlet free-form deformations and their application to hand simulation. In *Proc. Computer Animation*, pages 93–102, 1997.
24. M. Proesmans and L. Van Gool. Reading between the lines - a method for extracting dynamic 3d with texture. In *Proceedings of VRST*, pages 95–102, 1997.

© The Eurographics Association and Blackwell Publishers 2000.